IN THE CLAIMS

The claims are provided below for the Examiner's convenience. No claim amendments are being made.

1. (Currently Amended) A computer-implemented method for a running parent process to collect exit information from a child process, the child process associated with a nonterminated parent process, the computer-implemented method comprising:

identifying a parent process associated with a defunct child process; and

modifying the parent process associated with the defunct child process, wherein modifying the parent process by providing a thread associated with the parent process that ~~allows the parent process to~~ collects exit information associated with the defunct child process.

2. (Original) The computer-implemented method of claim 1, wherein modifying the parent process comprises co-opting a thread associated with the parent process to receive exit information associated with the defunct child process.

3. (Original) The computer-implemented method of claim 1, wherein modifying the parent process comprises creating an agent thread inside the parent process to collect exit information associated with the defunct process.

4. (Original) The computer-implemented method of claim 1, wherein modifying the parent process comprises altering the parent process to invoke wait() or waitpid().

5. (Original) The computer-implemented method of claim 1, wherein a control criterion is used to determine whether to modify the parent process.

6. (Original) The computer-implemented method of claim 5, wherein the control criterion comprises determining whether the parent process is stopped.

7. (Original) The computer-implemented method of claim 6, wherein the control criterion comprises determining whether the defunct process is a newly defunct process, wherein the parent is modified only if it is determined that the defunct process is not a newly defunct process.

8. (Original) The computer-implemented method of claim 6, wherein the control criterion comprises determining that the parent process is not an initial process.

9. (Original) A computer-implemented method for reaping a defunct child process associated with a parent process, the computer-implemented method comprising:

identifying a defunct child process;

attaching an agent thread to a parent process to allow modification of the parent process, wherein the parent process is modified to reap the defunct child process.

10. (Original) The computer-implemented method of claim 9, wherein a control criterion is used to determine whether to modify the parent process.

11. (Original) The computer-implemented method of claim 10, wherein the control criterion comprises determining whether the parent process is stopped.

12. (Original) The computer-implemented method of claim 10, wherein the control criterion comprises determining whether the defunct process is a newly defunct process, wherein the parent is modified only if it is determined that the defunct process is not a newly defunct process.

13. (Original) The computer-implemented method of claim 10, wherein the control criterion comprises determining that the parent process is not an initial process.

14. (Original) The computer program product comprising computer code for a running parent process to collect exit information from a child process, the child process associated with a nonterminated parent process, the computer program product comprising:

computer code for identifying a parent process associated with a defunct child process; and

computer code for modifying the parent process associated with the defunct child process, wherein modifying the parent process allows the parent process to collect exit information associated with the defunct child process.

15. (Original) The computer program product of claim 14, wherein modifying the parent process comprises co-opting a thread associated with the parent process to receive exit information associated with the defunct child process.

16. (Original) The computer program product of claim 14, wherein modifying the parent process comprises creating an agent thread inside the parent process to collect exit information associated with the defunct process.

17. (Original) An apparatus for a running parent process to collect exit information from a child process, the child process associated with a nonterminated parent process, the apparatus comprising:

means for identifying a parent process associated with a defunct child process; and

means for modifying the parent process associated with the defunct child process, wherein modifying the parent process allows the parent process to collect exit information associated with the defunct child process.

18. (Original) The apparatus of claim 17, wherein modifying the parent process comprises co-opting a thread associated with the parent process to receive exit information associated with the defunct child process.

19. (Original) The apparatus of claim 17, wherein modifying the parent process comprises creating an agent thread inside the parent process to collect exit information associated with the defunct process.

20. (Original) An apparatus for reaping a defunct child process associated with a parent process, the apparatus comprising:

memory;

a processor coupled to memory, the processor configured to identify a defunct child process and attach an agent thread to a parent process to allow modification of the parent process, wherein the parent process is modified to reap the defunct child process.

21. (Original) The apparatus of claim 20, wherein a control criterion is used to determine whether to modify the parent process.

22. (Original) The apparatus of claim 21, wherein the control criterion comprises determining whether the parent process is stopped.

23. (Original) The apparatus of claim 21, wherein the control criterion comprises determining whether the defunct process is a newly defunct process, wherein the parent is modified only if it is determined that the defunct process is not a newly defunct process.

24. (Original) The apparatus of claim 21, wherein the control criterion comprises determining that the parent process is not an initial process.